DIDLTools - Overview

Table of contents

1. What is a DIDL?

In the MPEG-21 Framework, complex digital objects are declared using the Digital Item Declaration Language (DIDL). DIDL introduces a set of abstract concepts that, together, form a well-defined data model for complex digital objects. Based on those abstract concepts, DIDL defines a W3C XML Schema that provides broad flexibility and extensibility for the actual representation of compliant complex digital objects.

The DIDL data model recognizes the following entities, which are visually represented in Figure 1:

- An **Item** is a grouping of **Items** and/or **Components**. In the XML representation, an **Item** is accommodated by the didl:Item element ({1} in Figure 1)
- A **Component** is a grouping of **Resources**. Multiple **Resources** in the same **Component** are considered equivalent and consequently an agent may use any one of them. In the XML representation, a **Component** is accommodated by the didl:Component element ({2} in Figure 1)
- A **Resource** is an individual datastream. In the XML Schema, a **Resource** is accommodated by the didl:Resource element ({3} in Figure 1)
- Secondary information pertaining to a **Container**, an **Item**, or a **Component** can be conveyed by means of a **Descriptor**. In the XML representation, a **Descriptor** is accommodated by the didl:Descriptor element ({4} in Figure 1). By definition, a didl:Descriptor is associated with its parent element in the XML representation. For example, a didl:Descriptor provided as a child element of a didl:Component is associated with that didl:Component.

Figure 1

2. What is the DIDL Toolkit?

DIDLTools is a Java toolkit for the construction, validation, serialization and de-serialization for MPEG-21 DIDL data model. DIDL, the MPEG-21 Digital Item Declaration Language, provides a hierarchical structure which allows each digital item to be modeled in the most appropriate fashion. The DID API allows for the construction of customized DID models, as well as provides flexible and extensible serialization methods.
Key Interfaces:

- Content Type - a simple interface for the addition of new DIDL Statement Content Types (i.e. DII, DCTERMS, XML-Signatures, etc.)
- Serialization - interface for the serialization of DID object objects and constituent content types and resource
- De-serialization - interface for the de-serialization of DID object objects and constituent

content types and resource

3. Limitations

The following elements in the DID Model are not supported by the DID API:

- Anchor
- Annotation
- Choice
- Components as child of Descriptor
- Condition
- Container

4. Additional Information

Bekaert, J., Van de Sompel, H. (2005, August).
Representing Digital Assets using MPEG-21 Digital Item Declaration
Bekaert, J., Liu, X., Van de Sompel, H. (2005, November).
Representing Digital Assets for Long-Term Preservation using MPEG-21 DID
Bekaert, J., Van de Sompel, H. (2005, June).
A Standards-based Solution for the Accurate Transfer of Digital Assets.
D-Lib Magazine, 11(6)